

Klasifikasi *Helpdesk* Menggunakan Metode *Support Vector Machine*

Stefanie Hilda Kusumahadi^{1*)}, Hartarto Junaedi², Joan Santoso³

¹Program Pascasarjana Teknologi Informasi, Sekolah Tinggi Teknik Surabaya, Surabaya

²Departemen Sistem Informasi, Sekolah Tinggi Teknik Surabaya, Surabaya

³Departemen Informatika, Sekolah Tinggi Teknik Surabaya, Surabaya

^{1,2,3}Jln. Ngagel Jaya Tengah 73-77, Surabaya, 60284, Indonesia

email: ¹stefaniehilda.jc@gmail.com, ²hartarto@stts.edu, ³joan@stts.edu

Copyright ©2019, Politeknik Harapan Bersama, Tegal

Abstract – The online helpdesk with ticketing system with the help of operators often experiences problems such as inappropriate delegation processes, the duration of the helpdesk waiting time to be delegated, even the helpdesk is missed to be handled. The ticket delegation checked manually by the operator has risks creating an error in delegating helpdesk tickets to inappropriate technicians. The helpdesk classification system is needed so that every incoming helpdesk ticket can be classified to the right technician according to the job description. The incoming Helpdesk is classified into 6 types of requests, namely multimedia, documentation, internet, server, hardware, software and miscellaneous. This helpdesk grouping is needed so that related technicians for each helpdesk can work and help the helpdesk according to their respective job descriptions. The Support Vector Machine method is used to classify text on the helpdesk. The use of Linear and Polynomial kernels produces an accuracy of 78%, the RBF or Gaussian kernel produces the highest accuracy of 81% while the Sigmoid kernel produces the smallest accuracy of 51%. The helpdesk classification results with the Support Vector Machine method can produce quite good accuracy.

Abstrak – *Helpdesk* secara online dengan sistem *ticketing* dengan bantuan operator sering kali mengalami permasalahan seperti proses pendelegasian yang kurang tepat, lamanya waktu tunggu *helpdesk* didelegasikan, bahkan terlewatnya *helpdesk* untuk dapat ditangani. Proses delegasi tiket secara manual oleh operator beresiko menimbulkan terjadinya kesalahan pendelegasian tiket *helpdesk* kepada teknisi yang tidak sesuai. Sistem klasifikasi *helpdesk* dibutuhkan agar setiap tiket *helpdesk* yang masuk dapat diklasifikasikan dan didelegasikan ke teknisi yang tepat sesuai dengan *job description*. *Helpdesk* yang masuk diklasifikasi menjadi 6 macam permintaan bantuan yaitu multimedia, dokumentasi, internet, server, hardware, software. Pengelompokan *helpdesk* ini diperlukan agar teknisi terkait untuk masing-masing *helpdesk* dapat mengerjakan dan membantu *helpdesk* sesuai dengan *job description* masing-masing. Metode *Support Vector Machine* dipakai untuk melakukan klasifikasi teks pada *helpdesk*. Penggunaan *kernel Linear* dan *Polynomial* menghasilkan akurasi sebesar 78%, *kernel RBF* atau *Gaussian* menghasilkan akurasi paling tinggi yaitu sebesar 81% sedangkan *kernel Sigmoid* menghasilkan akurasi paling kecil yaitu 51%. Hasil klasifikasi *helpdesk* dengan metode *Support Vector Machine* dapat menghasilkan akurasi cukup baik.

*) **Corresponding author:** (Stefanie Hilda Kusumahadi)

Email: stefaniehilda.jc@gmail.com

Kata Kunci – *Helpdesk*, *Support Vector Machine*, *Text Mining*, *Klasifikasi*.

I. PENDAHULUAN

Proses delegasi tiket secara manual oleh operator beresiko menimbulkan terjadinya kesalahan pendelegasian tiket *helpdesk* kepada teknisi yang tidak sesuai. Apabila operator *helpdesk* terlewat melakukan pengecekan setiap tiket, maka akan dapat terjadi tiket *helpdesk* tidak dilayani sehingga dapat menimbulkan komplain dari user. Kinerja teknisi menjadi kurang efisien apabila harus menunggu pendelegasian manual dari operator *helpdesk* yang juga bisa melakukan kesalahan serta dapat terjadi *delay*/penundaan pekerjaan karena proses pendelegasian yang masih manual [1]. Pelayanan departemen IT pada suatu organisasi menjadi bagian penting untuk tetap menjaga perangkat keras dan perangkat lunak yang digunakan perusahaan berjalan dengan [2]. Oleh karena itu sistem klasifikasi tiket *helpdesk* dapat membantu proses pendelegasian secara otomatis kepada teknisi terkait sehingga tidak lagi memerlukan bantuan operator. Penelitian ini dapat membantu mengurangi kesalahan pendelegasian tiket *helpdesk* kepada teknisi di Yayasan Pendidikan Generasi Rajawali.

Helpdesk Yayasan Pendidikan Generasi Rajawali digunakan sebagai *dataset* penelitian yang berjumlah 2401 tiket. Metode yang digunakan adalah *Support Vector Machine*. SVM telah dikenal sebagai salah satu metode *supervised machine learning* yang baik untuk klasifikasi teks [3].

SVM telah sering digunakan dalam klasifikasi teks karena memiliki performa yang cukup baik dibandingkan dengan metode lainnya seperti Neural Network dan Naive Bayes. Sebagai contoh pada penelitian Mucahit Altintas dan A. Cuneyd Tantug untuk klasifikasi tiket *helpdesk* didapat akurasi tertinggi menggunakan SVM yaitu sebesar 90%.

Pada penelitian ini dilakukan percobaan pada beberapa jenis *kernel* pada SVM untuk mengklasifikasikan tiket *helpdesk* yang belum ada pada penelitian terkait sebelumnya.

II. PENELITIAN YANG TERKAIT

Mucahit Altintas dan A. Cuneyd Tantug pada [1] melakukan penelitian terhadap tiket permintaan bantuan dengan sistem pendelegasian otomatis untuk mengurangi

waktu tunggu operator tiket dalam melakukan delegasi secara manual serta mengurangi kesalahan dalam pendelegasian. Dilakukan pengklasifikasian terhadap kategori tiket akan ditujukan kepada departemen yang mana, lalu diklasifikasikan kembali ke Sub kategori. Apabila prediksi memiliki angka *confidence* yang besar daripada *threshold value*, maka tiket akan diklasifikasikan pada kategori departemen dan sub kategori tersebut. Apabila nilai *confidence* tidak lebih besar dari *threshold value*, maka operator akan melakukan pendelegasian secara manual untuk tiket tersebut.

Klasifikasi pada penelitian ini menggunakan metode SVM dengan *Kernel Polynomial* karena *dataset non linearly separable*, *Naive Bayes*, *k-NN* dengan nilai $k=1$ dan *Decision Tree*. Hasil akurasi kategori departemen yang paling tinggi adalah metode SVM sebesar 90 % untuk model pembobotan TF-IDF dan *Boolean*, 88% untuk model pembobotan TF. Untuk sub kategori, pada departemen *Information Technology* dengan model pembobotan TF-IDF dan *classifier SVM* menghasilkan akurasi paling tinggi yaitu 80%, pada departemen *Student Affairs* dengan model pembobotan TF-IDF dan *classifier SVM* menghasilkan akurasi paling tinggi yaitu 80% juga. Untuk departemen *Health, Culture and Sport* model pembobotan TF-IDF dan *classifier Decision Tree* menghasilkan akurasi 93%. Pada departemen *Office of Scholarships and Dormitories*, *Naive Bayes* dan pembobotan TF menghasilkan akurasi paling tinggi yaitu 71%. Untuk tingkat akurasi tidak dijelaskan penggunaan *confusion matrix*.

Goby Niklas, Brandt Tobias, Feuerriegel Stefan, Neumann Dirk pada [2] melakukan penelitian tiket terkait permasalahan IT. *Text* pada tiket dilakukan *pre-processing* dan diubah menjadi bentuk *matrix* atau *document-term matrix*. *Matrix* ini yang nantinya menjadi *input* algoritma. Peneliti membagi *dataset* menjadi 80% *training set* dan 20% *testing set*. Akurasi rata-rata untuk seluruh topik sebesar 92.72%. sistem memiliki akurasi dan presisi lebih dari 85%.

Dingding Wang, Tao Li, Shenghuo Zhu, Yihong Gong pada [4] membuat sistem *helpdesk* pintar dengan mencari kesamaan *helpdesk* yang masuk dengan histori *helpdesk* yang pernah ada, dari *helpdesk* lama kemudian dipelajari dan digunakan sebagai acuan menyelesaikan *helpdesk* yang baru. Harapannya agar penyelesaian *helpdesk* dapat dilakukan lebih cepat karena sudah pernah menyelesaikan kasus serupa. Penelitian ini mencari kesamaan kasus dengan sistem *case-based* dan perangkikan. Pada penelitian pencarian kemiripan *helpdesk* dihasilkan akurasi sebesar 72%.

Support Vector Machine (SVM) adalah *classifier binary* yang dapat digunakan untuk klasifikasi teks multi kelas dengan memanfaatkan kernel. Adapun kelebihan SVM antara lain:

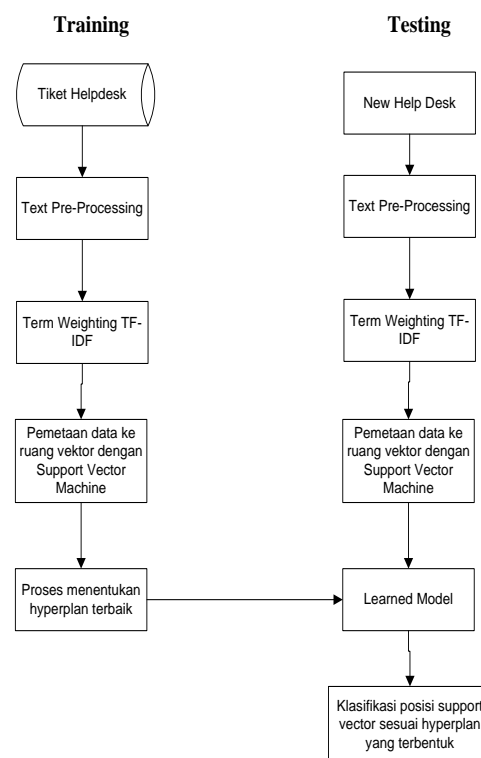
- Mampu mengolah data yang memiliki banyak fitur serta yang berdimensi tinggi dengan memanfaatkan fungsi Φ yang memetakan data ke ruang vektor berdimensi tinggi atau *high dimensional input space*.
- Merupakan Classifier yang memberikan solusi bersifat global optimal atau solusi yang sama untuk setiap percobaan. SVM tidak memberikan solusi bersifat local optimal yang hasilnya berbeda untuk setiap percobaan.

Sedangkan kekurangan SVM adalah untuk klasifikasi lebih dari 2 kelas harus melakukan percobaan dengan beberapa Kernel untuk mendapatkan performa dari Kernel terbaik.

Pada penelitian klasifikasi teks *helpdesk* dilakukan percobaan pada beberapa jenis *kernel SVM* antara lain *Kernel Linear*, *Kernel Polynomial*, *Kernel RBF*, dan *Kernel Sigmoid* untuk mengklasifikasikan tiket *helpdesk* yang belum ada pada penelitian terkait sebelumnya. Dilakukan percobaan pada masing-masing *kernel* dengan parameter berbeda yang kemudian dibandingkan performanya dengan tujuan memperoleh performa terbaik nilai parameter dari setiap *kernel*.

III. METODOLOGI PENELITIAN

Alur proses penelitian yang dilakukan adalah seperti pada Gbr 1.



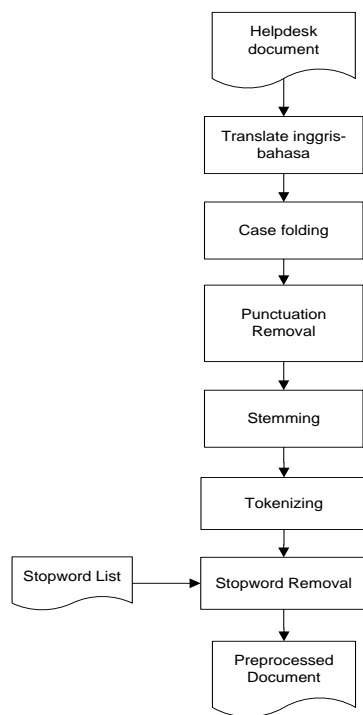
Gbr. 1 Alur Proses Klasifikasi Helpdesk

Proses klasifikasi tiket *helpdesk* dibagi menjadi 2 proses besar yaitu proses *training* dan proses *testing*. Pada proses *training* dan *testing*, dilakukan *pre-processing* terlebih dahulu pada tiket *helpdesk*. Setelah didapat kata-kata inti maka akan diberikan bobot dengan term weighting TF-IDF sehingga akan didapat nilai vector. Nilai vector kemudian diproses kedalam vector space. Support Vector Machine digunakan untuk mendapatkan hyperplane terbaik pada masing-masing kelas pada ruang vector.

A. Preprocessing Untuk Mendapatkan Daftar Kata Dari Tiket Helpdesk

Data *helpdesk* dari tahun 2014 hingga November 2018 dipakai menjadi *dataset* utama. Terlebih dahulu dilakukan *preprocessing* terhadap *dataset*. *Preprocessing* dilakukan 6 tahap. Pada proses *training* dan *testing*, dilakukan *pre-processing* terlebih dahulu pada tiket *helpdesk* antara lain melakukan pengecekan bahasa *helpdesk*, apabila berbahasa inggris maka diterjemahkan terlebih dahulu ke bahasa

indonesia, *case folding* atau mengubah huruf menjadi berukuran kecil, menghilangkan tanda baca dan angka, tokenisasi, *stop word removal* dan *stemming*. *Stemmer* sastrawi untuk bahasa Indonesia digunakan dalam proses *stemming*. *Text pre-processing* adalah tahapan penghilangan *noise* pada *text* sehingga menghasilkan *output* yang berupa *bag-of word* [5]. Tahap *preprocessing* ini meliputi:



Gbr. 2 Alur Preprocessing

1) *Case Folding*: pada tahap ini isi teks yang terdiri dari huruf capital dan huruf kecil atau *lower case* diubah agar menjadi huruf kecil keseluruhan. Hal ini dilakukan agar teks menjadi seragam.

2) *Tokenizing*: pada tahap ini dilakukan pemotongan string berdasarkan tiap kata. Untuk karakter selain alfabet atau karakter tanda baca seperti *whitespace*, *enter*, tabulasi, spasi, petik tunggal (‘), titik (.), semikolon (;), titik dua (:) dianggap sebagai pemisah kata. Selain itu tokenisasi juga membuang karakter-karakter tanda baca atau nonalfanumerik sehingga tersisa hanya sekumpulan kata-kata dari dalam dokumen.[6]

3) *Stop Word Removal*: merupakan tahap *filtering* yang adalah mengambil kata-kata penting dari hasil token serta membuang *common word*. Pada penelitian ini library stopword yang digunakan adalah Sastrawi. Algoritma *stoplist* (membuang kata kurang penting) atau *wordlist* (menyimpan kata penting). *Stoplist/stopword* adalah kata yang tidak deskriptif yang bisa jadi berupa kata penghubung yang tidak mendeskripsikan isi dari teks. Contoh dari *stopwords* adalah “yang”, “dan”, “di”, “dari” dan seterusnya.

4) *Stemming*: tahap ini adalah proses mengubah kata yang berimbuhan menjadi kata dasar. *Stemmer* Sastrawi digunakan dalam penelitian ini untuk menghilangkan awalan dan akhiran. Proses stemming pada teks yang berbahasa Indonesia berbeda dengan proses stemming pada teks berbahasa Inggris karena teks berbahasa indonesia bisa

mengandung awalan ataupun akhiran sedangkan teks berbahasa Inggris hanya ditambahkan akhhiran. Teks berbahasa Inggris diproses agar dapat menghilangkan sufiks. Sedangkan pada teks berbahasa Indonesia semua kata imbuhan baik itu sufiks dan prefiks juga dihilangkan pada proses stemming.

TABEL I
HASIL PREPROCESSING *HELPDESK*

Original Request Detail	Case Folding	Stemming	Tokenizing	Stop Word Removal
Tinta Printer Epson 850 yang buat print report. Yang perlu di refill, semua warna, soalnya sisa sedikit masing-masing warna. Thank you	tinta printer Epson 850 yang buat print report. yang perlu di refill, semua warna, soalnya sisa sedikit masing-masing warna. thank you	tinta printer Epson 850 yang buat print report yang perlu di refill semua warna soal sisa sedikit masing-masing warna. thank you	tinta printer Epson 850 yang buat print report yang perlu di refill semua warna soal sisa sedikit masing-masing warna thank you	tinta printer Epson print report refill warna warna
Mohon disediakan voucher internet untuk hari ini (31 Oct 2018) dari jam 11.00 - 14.00 untuk 26 users. Digunakan untuk pembelajaran matematika.	mohon disediakan voucher internet untuk hari ini (31 oct 2018) dari jam 11.00 - 14.00 untuk 26 users. digunakan untuk pembelajaran matematika.	mohon sedia voucher internet untuk hari ini 31 oct 2018 dari jam 11 00 -14 00 untuk 26 users guna untuk ajar matematika	mohon sedia internet untuk hari ini oct dari jam untuk users guna untuk ajar matematika	sedia voucher internet
MOHON DIBANTU UNTUK DI ISI TINTA PRINTER HRD	mohon dibantu untuk di isi tinta printer hrd	mohon bantu untuk di refill tinta printer hrd	mohon bantu untuk di isi tinta printer hrd	isi tinta printer hrd

Tabel I menunjukkan hasil dari *preprocessing* terhadap *text helpdesk*. Hasil akhir yaitu pada kolom *stop word removal* akan diolah ke proses selanjutnya yaitu *term weighting* dengan menggunakan TF-IDF.

B. Ekstraksi Fitur Hasil Preprocessing Menjadi Matriks dengan TF-IDF

TF-IDF (*Term Frequency Inverse Document Frequency*) merupakan algoritma yang digunakan untuk menghitung bobot

atau nilai frekuensi sebuah kata atau *term* di dalam sebuah dokumen dan juga dapat digunakan untuk menghitung bobot atau frekuensi di dalam banyak dokumen. Dalam model ruang *vector*, bobot dari *term* yang ada pada dokumen direpresentasikan oleh *term-document matrix* (atau *term-frequency matrix*) [7].

$$\begin{bmatrix} & T_1 & T_2 & \dots & T_t \\ D_1 & w_{11} & w_{21} & \dots & w_{t1} \\ D_2 & w_{12} & w_{22} & \dots & w_{t2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ D_n & w_{1n} & w_{2n} & \dots & w_{tn} \end{bmatrix}$$

Gbr 3. Contoh Matriks Term dari n - dokumen dan sejumlah t-term

Frekuensi kemunculan suatu term pada suatu dokumen disebut *Term Frequency* (TF) dengan formula seperti pada (1)

$$Tf = \begin{cases} 1 + \log_{10}(f_{t,d}), & f_{t,d} > 0 \\ 0, & f_{t,d} = 0 \end{cases} \quad (1)$$

Inverse Document Frequency (IDF) adalah jumlah dokumen dalam korpus dibagi jumlah dokumen yang mengandung term. Formula *Inverse Document Frequency* adalah seperti pada (2)

$$Idf = \log\left(\frac{D}{df_j}\right) \quad (2)$$

Sehingga formula TF-IDF adalah penggabungan formula *Term Frequency* dengan *Inverse Document Frequency* dengan cara mengkalikan seperti pada (3)

$$w_{ij} = Tf * Idf \quad (3)$$

C. Klasifikasi Text Menggunakan Support Vector Machine

Support Vector Machine adalah suatu algoritma yang dikembangkan oleh Boser, Guyon, Vapnik. Algoritma ini untuk pertama kalinya dipresentasikan ke publik pada tahun 1992 di *Annual Workshop on Computational Learning Theory*. Konsep dasar dari *Support Vector Machine* merupakan kombinasi dari teori-teori komputasi yang telah ada sebelumnya. Teori yang ada pada *Support Vector Machine* antara lain seperti margin *hyperplane*, *kernel* yang oleh Aronszajn dipresentasikan pada tahun 1950, demikian juga dengan konsep-konsep pendukung lainnya dari *Support Vector Machine*. [3] Algoritma SVM dapat menemukan *hyperplane* terbaik pada input space. *Hyperplane* adalah bidang pemisah suatu kelas dengan kelas lainnya. Prinsip dasar SVM adalah *linear classifier* yang berarti hanya bisa digunakan untuk mengklasifikasi data antara 2 kelas namun karena kasus pada dunia nyata umumnya adalah lebih dari 2 kelas maka dikembangkan lebih lanjut agar dapat bekerja pada masalah *non-linear* atau data *non-linear* dengan memasukkan konsep *kernel trick* dan menggunakan fungsi Φ agar dapat memetakan data kedalam ruang berdimensi tinggi.

Pada penelitian ini library yang digunakan untuk implementasi metode *Support Vector Machine* adalah Library Linear SVC (*Support Vector Classification*) dan One Versus Rest. One Versus Rest adalah metode yang digunakan untuk

multi klasifikasi yaitu klasifikasi lebih dari 2 kelas. Klasifikasi adalah suatu proses untuk menemukan model atau classifier yang memiliki performa terbaik sehingga mampu membedakan kelas dari data yang kemudian dilakukan prediksi kelas yang tidak diketahui dari suatu data lainnya. [8]

Masalah dalam dunia nyata umumnya bersifat *non-linear separable* atau tidak dapat dipisahkan secara linear oleh karena sebaran data yang beragam. Untuk menyelesaikan masalah *non linearly separable problem*, modifikasi metode SVM dilakukan dengan memasukkan fungsi *Kernel*. [9] [10]

1) Fungsi Kernel Linear

$$K(x_i, x_j) = (x_i^t x_j) \quad (4)$$

2) Fungsi Kernel Polynomial

$$K(x_i, x_j) = [(x_i \cdot x_j) + 1]^d \quad (5)$$

3) Fungsi Kernel Radial Basis Function (RBF)

$$K(x_i, x_j) = \exp(-\|x - x_i\|^2 / 2\sigma^2) \quad (6)$$

4) Fungsi Kernel Sigmoid

$$K(x_i, x_j) = \tanh(\beta_0 x_i \cdot x_j + \beta_1) \quad (7)$$

Kernel trick memberikan berbagai kemudahan untuk memecahkan permasalahan *non-linearly separable problem*. Pada proses pembelajaran SVM, untuk menentukan *support vector*, cukup dengan melakukan percobaan pada beberapa fungsi *kernel*. Selanjutnya performa antara kernel dibandingkan agar didapatkan *kernel* dengan performa terbaik. Model yang telah dilatih akan melakukan klasifikasi terhadap data testing berdasarkan pengetahuan yang telah dipelajari sebelumnya dari proses *training*.

D. Library Penunjang Aplikasi

Pada penelitian ini menggunakan beberapa *library* pendukung dalam pembuatan aplikasi. Adapun detail kegunaan tiap *library* [11], [12] yang dipakai dijelaskan dalam tabel II berikut.

TABEL II
LIBRARY YANG DIGUNAKAN DALAM PENELITIAN

No	Nama Library	Fungsi
1	sklearn.svm.LinearSVC	library Linear Support Vector Classification
2	sklearn.model_selection.Train_test_spilt	library untuk memisahkan array atau matrix menjadi train dan test set
3	sklearn.metrics.classification_report	library evaluasi classifier dengan output precision, recall, f1 score dan support
4	sklearn.metrics.confusion_matrix	library evaluasi classifier dengan output confusion matrix

TABEL III
LIBRARY YANG DIGUNAKAN DALAM PENELITIAN

No	Nama Library	Fungsi
5	sklearn.TfidfVectorizer	library untuk mengubah raw document menjadi matrix dari TF-IDF
6	sklearn.multiclass.OneVsRestClassifier	klasifikasi multiclass atau multilabel dan strateginya terdiri dari pemasangan satu classifier per kelas. Untuk setiap classifier, kelas tersebut dipasang terhadap semua kelas lainnya.
7	sklearn.preprocessing.label_binarize	library untuk mengubah kelas ke dalam bentuk binary
8	Natural Language Toolkit (NLTK)	sebuah platform yang digunakan untuk membangun program analisis teks.
9	Sastrawi	untuk melakukan stemming yaitu menghilangkan prefiks dan sufiks pada kata yang berimbuhan sehingga kembali ke bentuk dasarnya.

IV. HASIL DAN PEMBAHASAN

A. Skema Penelitian

Dataset helpdesk yang digunakan berjumlah total 2401 tiket. Tiket ini dari tahun 2014 hingga Oktober 2018. 70% atau sebanyak 1680 tiket digunakan sebagai *dataset* training. 30% atau 721 tiket digunakan untuk *testing*. Penelitian menggunakan *scikit learn* dengan *libsvm* sebagai tools yang dapat melakukan klasifikasi multi kelas dengan algoritma *one-against-one*. *Scikit learn* adalah modul berbasis Python yang mengintegrasikan *machine learning* algoritma permasalahan *supervised* dan *unsupervised*. [11]

Perhitungan performa dari suatu classifier dilakukan dengan menggunakan *confusion matrix*. Dalam penelitian ini, *Confusion matrix* adalah alat yang berguna untuk menganalisis seberapa baik classifier dalam melakukan klasifikasi. [13]

TABEL IV
DATASET HELPDESK

No	Tipe	Total
1	Multimedia	751
2	Dokumentasi	584
3	Server	41
4	Internet	110
5	Hardware	713
6	Software	202
Total Dataset Training		2401

TP (*true positif*) dan TN (*true negatif*) memberikan informasi ketika *classifier* melakukan klasifikasi dengan benar, sedangkan FP (*false positif*) dan FN (*false negatif*) memberitahu ketika *classifier* melakukan kesalahan dalam klasifikasi. Dari *confusion matrix* didapatkan nilai :

$$\text{Precision} = \frac{TP}{TP+FP} \quad (8)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (9)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

$$\text{F1 Measure} = 2 \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (11)$$

Tabel IV, V, VI, VII, VIII menunjukkan penggunaan *confusion matrix* sehingga didapatkan nilai *precision*, *recall*, *accuracy*, *F1 Score* dari masing-masing percobaan.

B. Hasil Percobaan

Percobaan pada tiap *kernel* yang berbeda dilakukan dengan menggunakan parameter dari angka terkecil hingga terbesar hingga didapat akurasi yang terbaik. Nilai parameter mengacu dari beberapa penelitian terkait yaitu [8][14].

Masing-masing kelas yaitu dokumentasi, *hardware*, *Internet*, *Multimedia*, *Server*, *Software* dihitung *precision* dan *recall*. Detil klasifikasi masing-masing kelas pada *kernel linear* adalah seperti pada tabel V.

TABEL V
HASIL TESTING KERNEL LINEAR

Kernel Linear	Precision	Recall	F1-Score	Support
Dokumentasi	0.77	0.74	0.76	173
Hardware	0.77	0.87	0.82	208
Internet	0.91	0.54	0.68	37
Multimedia	0.78	0.92	0.84	236
Server	1	0.17	0.29	12
Software	0.71	0.22	0.33	55

Percobaan menggunakan *kernel linear* menghasilkan akurasi sebesar 78% dengan nilai parameter terbaik yaitu C = 1. Kelas *server* memiliki nilai *precision* terbaik yaitu 1 namun nilai *recall* nya paling rendah yaitu 0.17. Kelas *multimedia*

memiliki *precision* 0.78, dengan *recall* tertinggi sebesar 0.92 dan *F1 score* tertinggi yaitu 0.84.

TABEL VI
HASIL TESTING KERNEL POLYNOMIAL

Kernel Polynomial	Precision	Recall	F1-Score	Support
Dokumentasi	0.77	0.77	0.77	175
Hardware	0.82	0.83	0.83	235
Internet	0.83	0.76	0.79	25
Multimedia	0.82	0.85	0.84	219
Server	0.57	0.44	0.50	9
Software	0.50	0.41	0.45	58

Percobaan menggunakan *kernel polynomial* menghasilkan akurasi sebesar 78% dengan nilai parameter terbaik yaitu $C = 1$, nilai r dan d adalah 10. Kelas *internet* memiliki *precision* tertinggi yaitu 0.83 dengan *recall* 0.76. kelas *multimedia* memiliki *precision* 0.82 dengan *recall* tertinggi yaitu 0.85 dan *F1 score* tertinggi 0.84.

TABEL VII
HASIL TESTING KERNEL RBF

Kernel RBF	Precision	Recall	F1-Score	Support
Dokumentasi	0.83	0.88	0.85	155
Hardware	0.78	0.89	0.83	215
Internet	1	0.44	0.61	41
Multimedia	0.83	0.89	0.86	239
Server	1	0.38	0.55	8
Software	0.74	0.41	0.53	63

Percobaan menggunakan *kernel RBF* atau *Gaussian* menghasilkan akurasi sebesar 81% dengan nilai parameter terbaik yaitu $C = 1$ dan $\gamma=10$. Kelas *internet* memiliki nilai *precision* tertinggi sebesar 1 dengan *recall* 0.44. kelas *multimedia* memiliki nilai *precision* sebesar 0.83 dan *recall* tertinggi yaitu 0.89 dan *F1 score* sebesar 0.86.

TABEL VIII
HASIL TESTING KERNEL SIGMOID

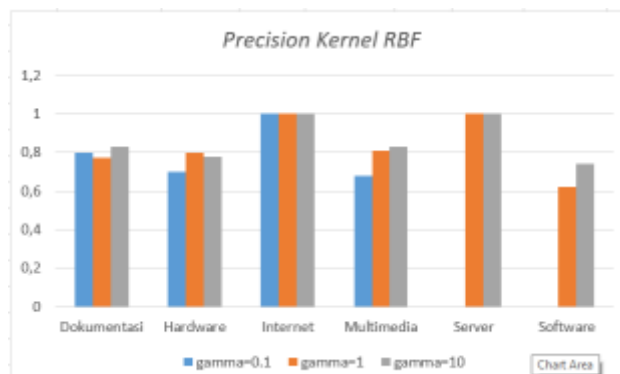
Kernel Sigmoid	Precision	Recall	F1-Score	Support
Dokumentasi	0	0	0	189
Hardware	0.80	0.74	0.77	213
Internet	0	0	0	29
Multimedia	0.40	0.94	0.56	222
Server	0	0	0	7
Software	0	0	0	61

Percobaan menggunakan *kernel Sigmoid* menghasilkan akurasi sebesar 51% dengan nilai parameter terbaik yaitu $C = 1$, $r=0.01$. kelas *hardware* memiliki nilai *precision* sebesar 0.80 dan *recall* 0.74 serta *F1 score* 0.77.

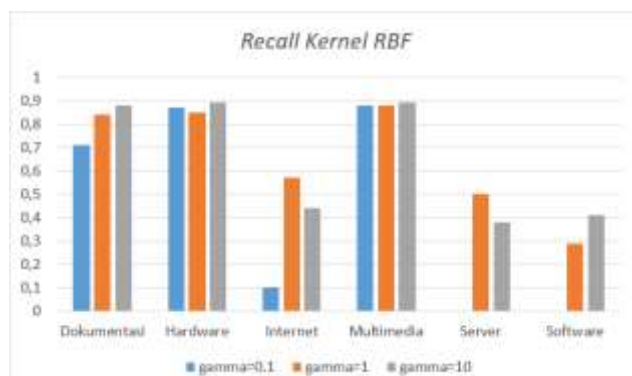
Kernel RBF atau *Gaussian* dapat memberikan akurasi tertinggi yaitu sebesar 81%. Perbandingan akurasi dari masing-masing *kernel* seperti pada tabel IX.

TABEL IX
AKURASI TERTINGGI MASING-MASING KERNEL DAN PARAMETERNYA

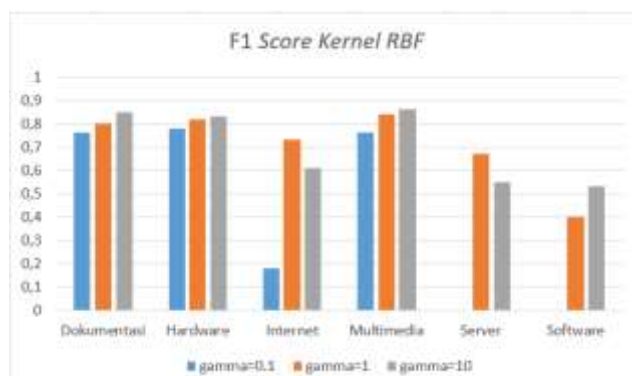
Kernel Function	Optimization Parameter				Akurasi
	C	γ	r	d	
Linear	1	-	-	-	0.78
RBF / Gaussian	1	10	-	-	0.81
Polynomial	1	-	10	10	0.78
Sigmoid	1	-	0,01	-	0.51



Gbr. 3 Perbandingan nilai *precision* percobaan nilai parameter kernel RBF/Gaussian



Gbr. 4 Perbandingan nilai *recall* percobaan nilai parameter kernel RBF/Gaussian



Gbr.5 Perbandingan nilai *F1 score* percobaan nilai parameter kernel RBF/Gaussian

Perbandingan nilai parameter *kernel RBF* atau *Gaussian* pada gambar 3 menunjukkan perbandingan nilai *precision* dari parameter *Gamma* dengan nilai 0,1, 1 dan 10. Nilai *Gamma*=10 memberikan nilai *precision* lebih baik dari pada 0,1 atau 1.

Perbandingan nilai *recall* dan *F1 score* pada nilai parameter $\text{Gamma}=10$ memberikan nilai yang lebih baik daripada nilai 0.1 atau 1.

V. KESIMPULAN

Metode *Support Vector Machine* dapat digunakan untuk melakukan klasifikasi tiket *helpdesk*. Penggunaan *kernel Linear* dengan nilai parameter $C = 1$ menghasilkan akurasi sebesar 78%. *Kernel Polynomial* dengan nilai parameter $C=1$, $r=10$, $d=10$ dapat menghasilkan akurasi sebesar 78%. Penggunaan *kernel Linear* dan *kernel Polynomial* menghasilkan akurasi yang sama. Parameter C secara *default* pada fungsi *kernel library scikit* bernilai 1 dan dapat memberi akurasi tinggi.

Kernel RBF atau *Gaussian* menghasilkan akurasi paling tinggi yaitu sebesar 81% dengan parameter $C=1$ dan $\gamma=10$. *Kernel sigmoid* menghasilkan akurasi paling rendah yaitu sebesar 51% dengan parameter $C=1$ dan $r=0.01$. *Kernel Gaussian* dengan parameter $\text{Gamma}=10$ menghasilkan akurasi paling tinggi dibandingkan dengan parameter Gamma lainnya.

Pada penelitian mendatang, akan dicoba untuk menerapkan algoritma *Latent Semantic Indexing* pada output TF-IDF sehingga proses ekstraksi fitur dapat menghasilkan fitur yang lebih baik untuk kemudian dipakai pada Model *Support Vector Machine*. Untuk mendapatkan akurasi lebih tinggi juga dapat dilakukan penyaringan kata terhadap *bag of word* dengan menambah daftar *stopword* serta penggunaan *cross validation*[15] dimana dataset dibagi menjadi *test set*, *train set* dan *validation set*.

UCAPAN TERIMA KASIH

Ucapan terima kasih penulis tujuikan kepada dosen pembimbing dan dosen co pembimbing Sekolah Tinggi Teknik Surabaya serta rekan-rekan di Yayasan Pendidikan Generasi Rajawali yang telah memberikan dukungan dalam bentuk saran, pengetahuan dan data hingga terselesaikannya penelitian ini.

DAFTAR PUSTAKA

- [1] M. Altintas and A. C. Tantug, "Machine Learning Based Tiket Classification in Issue Tracking Systems," *Proceeding Int. Conf. Artif. Intell. Comput. Sci. (AICS 2014)*, no. September, pp. 195–207, 2014.
- [2] N. Goby, T. Brandt, S. Feuerriegel, D. Neumann, and C. Research Goby, "Business Intelligence for Business Processes: the Case of It Incident Management," *ECIS Proc.*, no. April, pp. 1–15, 2016.
- [3] I. Pilászy, "Text Categorization and Support Vector Machines," vol. 1, pp. 1–10, 2004.
- [4] D. Wang, T. Li, S. Zhu, and Y. Gong, "iHelp: An intelligent online helpdesk system," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 41, no. 1, pp. 173–182, 2011.
- [5] Informatikalogi, "Text Preprocessing," 2016. [Online]. Available: <https://informatikalogi.com/text-preprocessing/>. [Accessed: 05-Dec-2018].
- [6] J. Santoso *et al.*, "Self-Training Naive Bayes Berbasis Word2Vec untuk Kategorisasi Berita Bahasa Indonesia," *JNTETI*, vol. 7, no. 2, pp. 158–166, 2018.
- [7] K. J. Cios, W. Pedrycz, R. W. Swiniarski, and L. A. Kurgan, *Data Mining A Knowledge Discovery Approach*, vol. 30, no. 11, 2007.
- [8] R. Munawarah, O. Soesanto, and M. R. Faisal, "Penerapan Metode Support Vector Machine Pada Diagnosa Hepatitis," *Kumpul. J. Ilmu Komput.*, vol. 04, no. 01, pp. 103–113, 2013.
- [9] A. S. Nugroho, A. B. Witarto, and D. Handoko, "Support Vector Machine , Teori dan Aplikasinya dalam Bioinformatika," *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 3, no. 3, pp. 204–215, 2011.
- [10] S. Lee and H. BYUN, "A survey on pattern recognition applications of support vector machines," *Int. J. Pattern Recognit.*, vol. 17, no. 3, pp. 459–486, 2003.
- [11] É. D. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, "Scikit-learn: Machine Learning in Python," 2011. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>.
- [12] V. Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, P. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, A. and and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, and E. Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, "Scikit Learn: Support Vector Machine." [Online]. Available: <https://scikit-learn.org/stable/modules/svm.html>. [Accessed: 14-Jan-2019].
- [13] J. Han, M. Kamber, and J. Pei, *Data Mining Concepts and Techniques*. 2011.
- [14] R. Diani, U. N. Wisesty, and A. Aditsania, "Analisis Pengaruh Kernel Support Vector Machine (SVM) pada Klasifikasi Data Microarray untuk Deteksi Kanker," *Ind. J. Comput.*, vol. 2, pp. 109–118, 2017.
- [15] Antoni Wibowo, "10 Fold-Cross Validation," 2017. [Online]. Available: <https://mti.binus.ac.id/2017/11/24/10-fold-cross-validation/>. [Accessed: 16-Nov-2018].